

PROGRAM DISPATCH PROCESSING SYSTEM

Publication number: JP5035506 (A)

Publication date: 1993-02-12

Inventor(s): KAYANO YUJI +

Applicant(s): FUJITSU LTD +

Classification:

- International: G06F9/46; G06F9/48; G06F9/46; (IPC1-7): G06F9/46

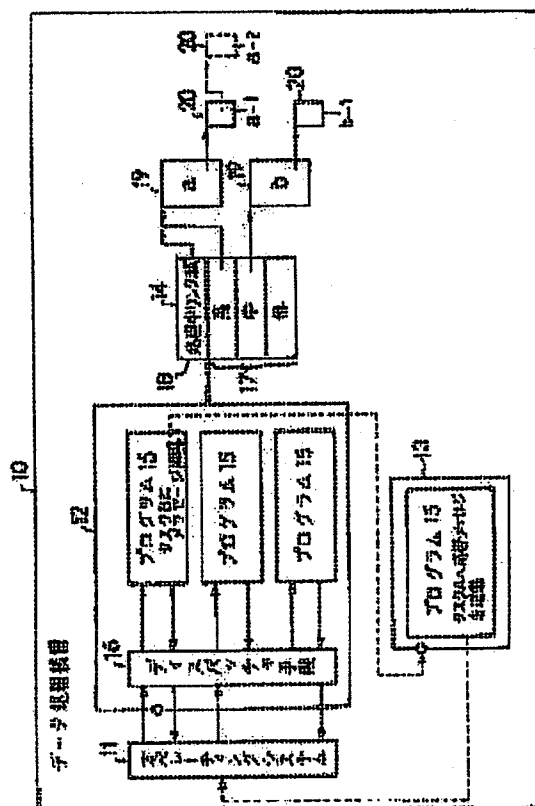
- European:

Application number: JP19910186025 19910725

Priority number(s): JP19910186025 19910725

Abstract of JP 5035506 (A)

PURPOSE: To surely dispatch the programs with preference to the data processing requests of the higher priority. **CONSTITUTION:** A data processor is provided with a request reception task 12 which accepts the data processing requests and an execution task 13 which receives the control of the task 12 and processes the data. A request queue terminal 14 is added to link the accepted data processing requests with each other according to the priority and also to control the data processing request under execution. A dispatcher means 16 which works with the task 12 carries out with preference the requests having higher priority than the request which is executed by the terminal 14 when the means 16 controls the data processing request carried out by the terminal 14. Thus these requests are controlled as the new data processing requests under execution.



Data supplied from the *espacenet* database — Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-35506

(43)公開日 平成5年(1993)2月12日

(51)Int.Cl.⁵

G O 6 F 9/46

識別記号

庁内整理番号

3 4 0 B 8120-5B

3 2 2 C 8120-5B

FI

技術表示箇所

審査請求 未請求 請求項の数 1 (全 8 頁)

(21)出願番号

特願平3-186025

(22)出願日

平成3年(1991)7月25日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 柏野 祐二

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74)代理人 弁理士 森田 寛 (外1名)

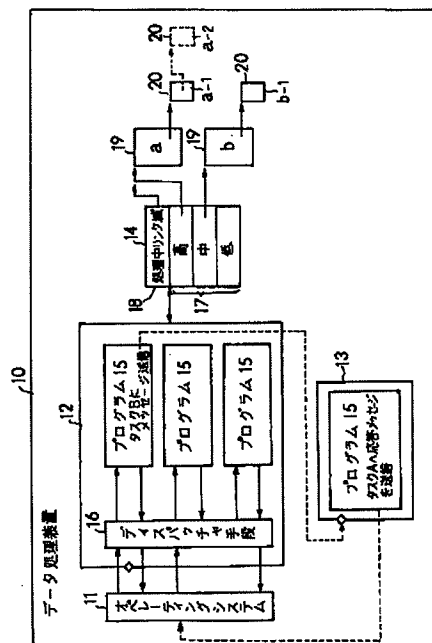
(54)【発明の名称】 プログラムデイスパッチ処理方式

(57) 【要約】

【目的】本発明は、複数タスクを制御するデータ処理装置においてのプログラムディスパッチ処理方式に関し、高い優先度のデータ処理要求に対してのプログラムを確実に優先的にディスパッチできるようにすることを目的とする。

【構成】データ処理要求を受け付ける要求受付タスク 12 と、要求受付タスク 12 から制御が渡されてデータ処理を実行するタスク 13 とを備えるデータ処理装置において、受け付けられたデータ処理要求を優先度別にリンクするとともに、実行中のデータ処理要求を管理する要求キューターミナル 14 を備え、要求受付タスク 12 上で動作するディスパッチャ手段 16 は、要求キューターミナル 14 が実行中のデータ処理要求を管理する場合には、そのデータ処理要求よりも高い優先度のデータ処理要求を優先的に実行して新たに実行中のデータ処理要求として管理していくように構成する。

本発明の原理構成図



【特許請求の範囲】

【請求項1】 データ処理要求を受け付ける要求受付タスク(12)と、該要求受付タスク(12)から制御が渡されてデータ処理を実行する1つ又は複数のタスク(13)とを備えるデータ処理装置において、上記要求受付タスク(12)により受け付けられたデータ処理要求を優先度別にリンクするとともに、実行中のデータ処理要求を管理する要求キューターミナル(14)を備え、上記要求受付タスク(12)上で動作するディスパッチャ手段(16)は、プログラム処理の終了時点において、上記要求キューターミナル(14)が実行中のデータ処理要求を管理するときに、該データ処理要求よりも高い優先度のデータ処理要求があるときには、その高いデータ処理要求の内の最も高い優先度のデータ処理要求に対応付けられるプログラムをディスパッチするとともに、該データ処理要求を実行中のものとして管理し、高いデータ処理要求がないときには、その実行中のデータ処理要求に対応付けられるプログラムをディスパッチし、一方、実行中のデータ処理要求を管理していないときには、最も高い優先度のデータ処理要求に対応付けられるプログラムをディスパッチするとともに、該データ処理要求を実行中のものとして管理していくよう処理することを、特徴とするプログラムディスパッチ処理方式。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、複数タスクを利用して1つのデータ処理要求を処理していく構成を採るデータ処理装置においてのプログラムディスパッチ処理方式に関し、特に、高い優先度のデータ処理要求に対してのプログラムを確実に優先的にディスパッチできるようにするプログラムディスパッチ処理方式に関するものである。

【0002】 オンラインシステム等に利用される計算機では、一般に、複数のタスクで複数のプログラムを一度に動作させることができる。このような計算機において、異常時のリカバリ処理要求のような緊急に処理すべきデータ処理要求が生ずる場合には、他の低い優先度のデータ処理要求よりも優先的に多くのタスクを割り当てたり、タスク毎に優先度を設けて高い優先度のタスクにその要求に対する処理をディスパッチしている。この場合、高い優先度のデータ処理要求に対してのプログラムが確実に優先的にディスパッチされる構成にしていく必要がある。

【0003】

【従来の技術】 従来では、タスク毎にデータ処理要求の優先度を管理する要求キューターミナルを備えて、タスク上で動作するディスパッチャプログラムが、優先度の高い順に、各データ処理要求を処理するプログラムをディスパッチしていく構成を採っていた。

【0004】 すなわち、図5に示すように、データ処理要求の優先度を管理する要求キューターミナル1を備えて、この要求キューターミナル1に、処理要求の制御表2をキューイングする構成を採る。そして、タスク上で動作するディスパッチャプログラム3は、図6の処理フローに示すように、優先度の高い順にキューを先頭から参照することで処理要求の制御表2を特定して、その特定した制御表2に対応するプログラムを呼び出していくとともに、キューのリンクを張り替えていくことで、優先度の高い順に各データ処理要求を処理するプログラムをディスパッチしていく構成を採っていたのである。この構成により、図5の例で説明するならば、からの順番の制御表2に対応付けられるプログラムが順番にディスパッチされていたのである。

【0005】

【発明が解決しようとする課題】 しかしながら、このような従来技術では、複数タスクを利用して1つのデータ処理要求を処理していく場合に、高い優先度のデータ処理要求に対してのプログラムが優先的にディスパッチされないことが起こるという問題点があった。

【0006】 すなわち、図7に示すように、高い優先度のデータ処理要求のプログラムが、タスクAで或るフェーズの処理を行った後にタスクBに制御を受け渡してタスクA上のディスパッチャプログラム3に復帰すると、このディスパッチャプログラム3は、次の優先度のデータ処理要求のプログラムをディスパッチしていくことになる。これから、タスクBの処理結果を元にして次のフェーズの高い優先度のデータ処理要求のプログラムをタスクAで実行しようとしても、低い優先度のデータ処理要求によりタスクAが占有されているために、その実行が遅れてしまうという問題点があったのである。

【0007】 本発明はかかる事情に鑑みてなされたものであって、複数タスクを利用して1つのデータ処理要求を処理していく構成を取るデータ処理装置において、高い優先度のデータ処理要求に対してのプログラムを確実に優先的にディスパッチできるようにする新たなプログラムディスパッチ処理方式の提供を目的とする。

【0008】

【課題を解決するための手段】 図1に本発明の原理構成を図示する。図中、10は本発明により構成されるデータ処理装置であって、オペレーティングシステム11と、データ処理要求を受け付ける要求受付タスク12と、要求受付タスク12から制御が渡されてデータ処理を実行する1つ又は複数のタスク13と、要求キューターミナル14とを備える。

【0009】 要求受付タスク12は、データ処理要求に応答する処理を実行する複数のプログラム15と、これらのプログラム15のディスパッチ処理を実行するディスパッチャ手段16とを展開する。要求キューターミナ

ル14は、要求受付タスク12により受け付けられたデータ処理要求を優先度別にリンクするリンク域17と、実行中のデータ処理要求をリンクする処理中リンク域18とを備える構成を採るものであって、このリンク域17は、データ処理要求を表示する要求テーブル19を優先度別にリンクし、処理中リンク域18は、データ処理中の要求テーブル19をリンクする。これらの要求テーブル19には、データ処理要求の処理依頼を表示する処理依頼制御表20がリンクされることになる。

【0010】要求受付タスク12は、外部から初めてデータ処理要求を受け付けると、要求テーブル19を作成して優先度別にリンク域17にリンクする。また、外部から初めてデータ処理要求を受け付けるとき、内部的に他タスク13と同期を取って要求受付タスク12で処理を実行するときに、処理依頼制御表20を作成して要求テーブル19にリンクする。

【0011】

【作用】本発明では、ディスパッチャ手段16は、プログラム処理の終了時点において、要求キュータミナル14の処理中リンク域18に要求テーブル19がリンクされているときに、その要求テーブル19よりも高い優先度の要求テーブル19があるときには、その高い要求テーブル19の内の最も高い優先度の要求テーブル19にリンクされる処理依頼制御表20を探して、その処理依頼制御表20の指すプログラム15を要求受付タスク12上で動作させるとともに、処理中リンク域18がその最も高い優先度の要求テーブル19をリンクするように設定する。そして、処理中リンク域18に要求テーブル19がリンクされているものの、その要求テーブル19よりも高い優先度の要求テーブル19がないときには、その要求テーブル19にリンクされる処理依頼制御表20を探して、その処理依頼制御表20の指すプログラム15を要求受付タスク12上で動作させる。

【0012】一方、処理中リンク域18に要求テーブル19がリンクされていないときには、要求キュータミナル14にリンクされる要求テーブル19を優先度順にサーチして、最も高い優先度の要求テーブル19を特定し、その要求テーブル19にリンクされる処理依頼制御表20を探して、その処理依頼制御表20の指すプログラム15を要求受付タスク12上で動作させるとともに、処理中リンク域18がその最も高い優先度の要求テーブル19をリンクするように設定する。このとき、処理依頼制御表20が見つからない場合には、ディスパッチャ手段16は、オペレーティングシステム11に復帰し、要求受付タスク12は、データ処理要求を受け取るまでの間何も動作しない状態に入る。

【0013】このようにして、起動されると、要求受付タスク12上で動作するプログラム15は、規定のデータ処理を実行し、必要に応じて、タスク13上で動作するプログラムを起動していく。一方、このようにして起

動されると、タスク13上で動作するプログラムは、必要に応じて、呼び出し元となる処理中リンク域18の要求テーブル19に新たな処理依頼制御表20をリンクしていく。

【0014】そして、要求受付タスク12上で動作するプログラム15は、プログラム処理を終了すると、ディスパッチャ手段16に対して、処理中リンク域18にリンクされる要求テーブル19にリンクされる最後の処理依頼制御表20の指すものであるか否かを通知するので、ディスパッチャ手段16は、この通知を受けて、最後の処理依頼制御表20の指すプログラム処理である場合には、その要求テーブル19とその処理依頼制御表20とを回収し、一方、最後の処理依頼制御表20の指すプログラム処理でない場合には、その処理依頼制御表20のみを回収していく。

【0015】この本発明の処理に従って、以下に説明するように、複数タスクを利用して1つのデータ処理要求を処理していく場合に、高い優先度のデータ処理要求に対してのプログラムが確実に優先的にディスパッチされることになる。以下、説明の便宜上、図1に示すように、処理中リンク域18にリンク（リンク域17の「高」にもリンク）される要求テーブル19を要求テーブルa、この要求テーブルaにリンクされる2つの処理依頼制御表20を処理依頼制御表a-1、a-2と表し、リンク域17の「中」にリンクされる要求テーブル19を要求テーブルb、この要求テーブルbにリンクされる処理依頼制御表20を処理依頼制御表b-1と表すものとする。

【0016】要求受付タスク12上で動作する処理依頼制御表a-1のプログラム15が、タスク13にメッセージで処理を依頼し、このメッセージを受けて、タスク13上で動作する対応のプログラムがデータ処理を実行して、処理依頼制御表a-2を処理依頼制御表a-1にリンクしていく。このとき、処理依頼制御表a-1のプログラム15は、プログラム処理を終了すると、上述の処理に従って、ディスパッチャ手段16に対して要求テーブルaの最後の処理依頼でないことを通知する。ディスパッチャ手段16は、この通知を受け取ると、要求テーブルaが要求テーブルbよりも優先度が高く、かつ要求テーブルaを処理中リンク域18から外す条件とはならないので、上述の処理に従って、要求テーブルbの処理依頼制御表b-1のプログラム15を要求受付タスク12上で動作させない。これから、タスク13で処理を終了して、その処理結果に従って要求受付タスク12で動作することになる処理依頼制御表a-2のプログラム15は、直ちに要求受付タスク12上で動作することができることになる。

【0017】この処理依頼制御表a-2のプログラム15は、プログラム処理を終了すると、上述の処理に従って、ディスパッチャ手段16に対して要求テーブルaの

最後の処理依頼であることを通知するので、ディスパッチ手段16は、要求テーブルbを処理中リンク域18にリンクして、処理依頼制御表b-1のプログラム15を要求受付タスク12上で動作させていく。このようにして、要求受付タスク12上では、処理依頼制御表a-1のプログラム15、処理依頼制御表a-2のプログラム15、処理依頼制御表b-1のプログラム15の順にディスパッチされていくことになる。

【0018】このように、本発明によれば、複数タスクを利用して1つのデータ処理要求を処理していく構成を取るデータ処理装置において、高い優先度のデータ処理要求に対してのプログラムを確実に優先的にディスパッチできるようにする。

【0019】

【実施例】以下、実施例に従って本発明を詳細に説明する。図2に示すように、複数タスクの制御と、そのタスク間でのメッセージによる制御のやり取りの可能であるオペレーティングシステムにおいて、タスクA～Dという4つのタスクを起動してデータ処理を実行するデータ処理装置を実施例として想定する。そして、この4つのタスクの内、タスクAを外部の端末からのコマンドを受け付けるタスク、その他のタスクB～Dをそれぞれが並列処理の可能な処理タスクであることを想定し、タスクAは、端末からのコマンドを受け付けて、タスクB～Dの内の対応するタスクに制御を渡し、この制御が渡されると、タスクB～Dは、規定のデータ処理を実行してタスクAに制御を戻し、タスクB～Dから制御が戻ると、タスクAは、コマンドの発行元の端末に対してコマンドの応答を返す処理を実行していくことを想定する。ここで、図2中、16aは、タスクA上で動作するディスパッチャプログラムである。

【0020】このようなデータ処理装置において、タスクAが低い優先度のコマンドの要求を受け付けた後に、異常が発生して緊急にその異常をリカバリするコマンドの要求を続けて受け付けるときに、本発明が有効なものとなる。すなわち、このような場合、本発明を実現するために備える図1で説明した要求キューミナル14は、図3に示すように、リンク域17の「中」にコマンドの要求テーブル19がリンクされるとともに、リンク域17の「高」にコマンドの要求テーブル19がリンクされる。そして、コマンドの要求テーブル19に、コマンドの処理依頼制御表20である処理依頼-1と、タスクBからの処理依頼制御表20である処理依頼-2とがリンクされるとともに、コマンドの要求テーブル19に、コマンドの処理依頼制御表20である処理依頼-1と、タスクCからの処理依頼制御表20である処理依頼-2とがリンクされることになる。

【0021】このコマンド／に対しての具体的な処理の説明に入る前に、図4の処理フローに従って、タスクA上で動作するディスパッチャプログラム16aの実

行する処理について詳細に説明する。

【0022】タスクA上で動作するディスパッチャプログラム16aは、プログラム処理の終了時点において、図4の処理フローに示すように、まず最初に、ステップ1で、処理中リンク域18に要求テーブル19がリンクされているか否かを判断する。この判断により、処理中リンク域18に何も要求テーブル19がリンクされていないと判断するときには、ステップ2に進んで、要求キューミナル14の全リンク域17にリンクされる要求テーブル19を優先度順にサーチして、最も高い優先度の要求テーブル19と、その要求テーブル19にリンクされる処理依頼制御表20とを見つける。一方、このステップ1の判断により、処理中リンク域18に要求テーブル19がリンクされていると判断するときには、ステップ3に進んで、処理中リンク域18にリンクされる要求テーブル19よりも高い優先度の要求テーブル19をサーチして、その高い要求テーブル19の内の最も高い優先度の要求テーブル19と、その要求テーブル19にリンクされる処理依頼制御表20とを見つける。このとき、処理中リンク域18にリンクされる要求テーブル19よりも高い優先度の要求テーブル19がない場合には、その処理中リンク域18にリンクされる要求テーブル19と、その要求テーブル19にリンクされる処理依頼制御表20とを見つけることになる。

【0023】ステップ2及びステップ3の処理に従って、要求テーブル19が見つからない場合、すなわち、処理依頼制御表20が見つからない場合には、ステップ4を介してそのまま処理を終了する。一方、要求テーブル19が見つかる場合には、ステップ4を介してステップ5に進んで、この見つけた要求テーブル19を処理中リンク域18にリンクする。そして、続くステップ6で、この見つけた処理依頼制御表20の指すタスクA上のプログラムを呼び出していく。このようにしてディスパッチャプログラム16aにより呼び出されると、タスクA上のプログラムは、タスクB～Dの内の対応するタスクに制御を渡し、この制御が渡されると、タスクB～Dは、規定のデータ処理を実行し、必要に応じて、呼び出し元となる処理中リンク域18の要求テーブル19に新たな処理依頼制御表20をリンクしていく。そして、タスクA上のプログラムは、プログラム処理を終了すると、ディスパッチャプログラム16aに対して、要求テーブル19にリンクされる最後の処理依頼制御表20の指すものであるか否かを通知してくるので、ディスパッチャプログラム16aは、ステップ7で、この通知を判断して、最後の処理依頼制御表20であることを判断するときには、ステップ8に進んで、その要求テーブル19とその処理依頼制御表20とを回収し、一方、最後の処理依頼制御表20でないことを判断するときには、ステップ9に進んで、その処理依頼制御表20のみを回収してから、ステップ1に戻るよう処理する。

【0024】次に、このディスパッチャプログラム16aの実行する処理に従って、図2及び図3に示した上述のコマンド／に対して実行されることになる処理について説明する。

【0025】タスクAのディスパッチャプログラム16aは、端末から低い優先度のコマンドを受け取ると、上述の処理に従って処理依頼-1の指すプログラムを起動する。このようにして起動されると、処理依頼-1の指すプログラムは、図2に示すように、その制御をメッセージでタスクBに渡し、この制御を受け取るタスクB上のプログラムは、コマンドの処理を実行して新たな処理依頼-2を生成していく。このとき、緊急を要する高い優先度のコマンドが未だ発行されていないので、処理中リンク域18は、コマンドの要求テーブル19をリンクしている。続いて、上述したように、緊急を要する高い優先度のコマンドが端末から発行されてくることになる。そして、処理依頼-1の指すプログラムは、ディスパッチャプログラム16aに対して、最後の処理依頼制御表20でないことを通知して復帰する。

【0026】ディスパッチャプログラム16aは、この復帰を受けて、上述の処理に従って、コマンドが処理中にも関わらず、コマンドの処理依頼-1を見つけて、その処理依頼-1の指すプログラムを起動するとともに、コマンドの要求テーブル19を処理中リンク域18にリンクする。このようにして起動されると、処理依頼-1の指すプログラムは、図2に示すように、その制御をメッセージでタスクCに渡し、この制御を受け取るタスクC上のプログラムは、コマンドの処理を実行して新たな処理依頼-2を生成していく。そして、処理依頼-1の指すプログラムは、ディスパッチャプログラム16aに対して、最後の処理依頼制御表20でないことを通知して復帰する。

【0027】一方、処理依頼-1の指すプログラムにより制御の渡されたタスクB上のプログラムは、コマンドの処理を終了するとタスクAに制御を戻していく。このとき、ディスパッチャプログラム16aは、処理中リンク域18がコマンドの要求テーブル19をリンクしていることから、上述の処理に従って、タスクBからコマンドの処理の終了通知を受け取っても、次の処理依頼-2の指すプログラムを呼び出さずにオペレーティングシステムに復帰していくことになる。

【0028】その後、処理依頼の指すプログラムにより制御の渡されたタスクC上のプログラムは、コマンドの処理を終了するとタスクAに制御を戻していくことになるが、このときには、ディスパッチャプログラム16aは、上述の処理に従って、直ちに処理依頼-2の指すプログラムを呼び出してタスクA上で実行する。この処理依頼-2の指すプログラムは、処理を終了すると、

ディスパッチャプログラム16aに対して、最後の処理依頼制御表20であることを通知して復帰する。この復帰を受けて、ディスパッチャプログラム16aは、上述の処理に従って、コマンドの要求テーブル19を処理中リンク域18にリンクするとともに、処理依頼-2の指すプログラムを呼び出してタスクA上で実行し、この処理依頼-2の指すプログラムは、処理を終了すると、ディスパッチャプログラム16aに対して、最後の処理依頼制御表20であることを通知して復帰する。

【0029】このようにして、高い優先度のコマンドのプログラムの方が、低い優先度のコマンドのプログラムよりもタスクA上で優先的に呼び出されて実行されていくことで、コマンドの要求の方が、コマンドの要求よりも先に処理を終了していくことになるのである。

【0030】

【発明の効果】以上説明したように、本発明によれば、複数タスクを利用して1つのデータ処理要求を処理していく構成を採るデータ処理装置において、高い優先度のデータ処理要求に対してのプログラムが確実に優先的にディスパッチされるようになる。これにより、低い優先度のデータ処理要求を受け付けた後に高い優先度のデータ処理要求を受け付けた場合に、高い優先度のデータ処理要求に対する処理が低い優先度の処理中に割り込まれて処理されることになって、高い優先度のデータ処理要求の方が先に処理を終了していくことになるのである。

【図面の簡単な説明】

【図1】本発明の原理構成図である。

【図2】本発明の一実施例である。

【図3】本発明の一実施例である。

【図4】ディスパッチャプログラムの実行する処理フローの一実施例である。

【図5】従来技術の説明図である。

【図6】従来技術の説明図である。

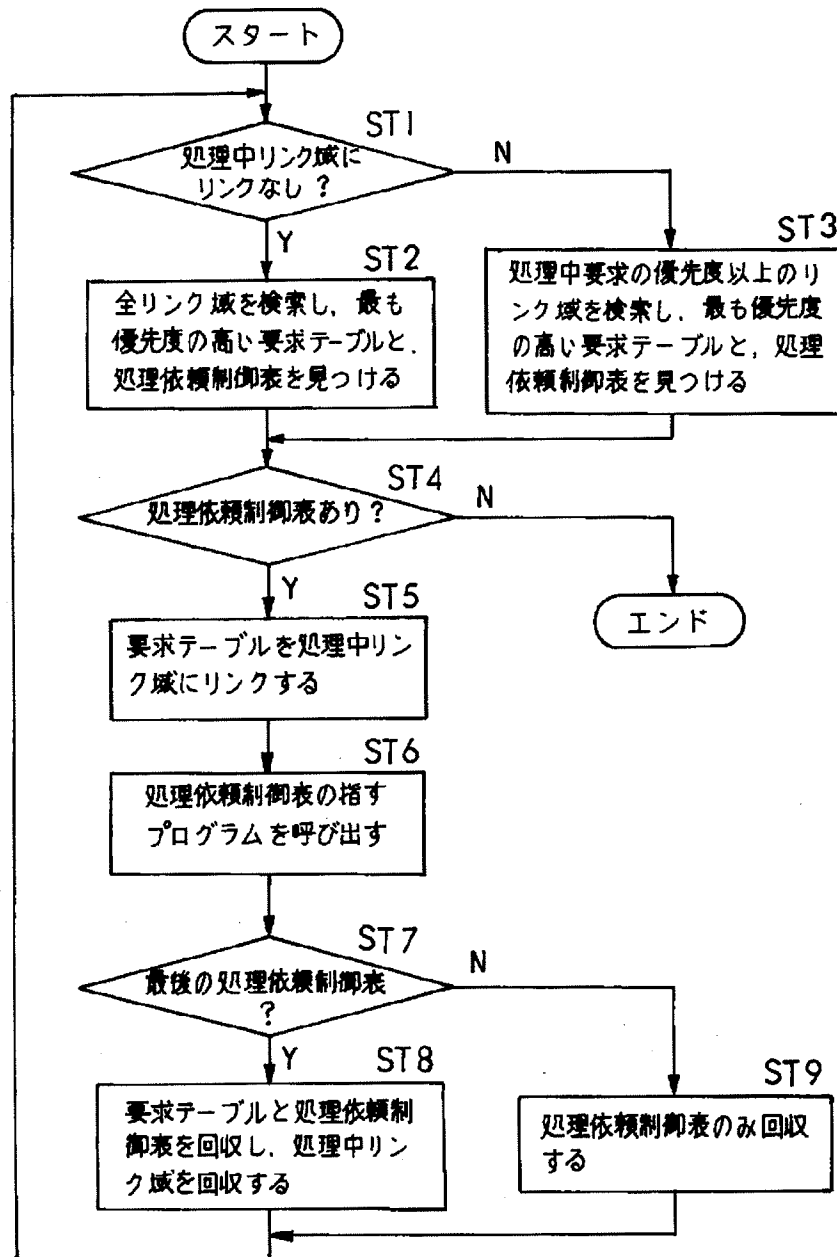
【図7】従来技術の問題点の説明図である。

【符号の説明】

- 10 データ処理装置
- 11 オペレーティングシステム
- 12 要求受付タスク
- 13 タスク
- 14 要求キューターミナル
- 15 プログラム
- 16 ディスパッチャ手段
- 17 リンク域
- 18 処理中リンク域
- 19 要求テーブル
- 20 処理依頼制御表

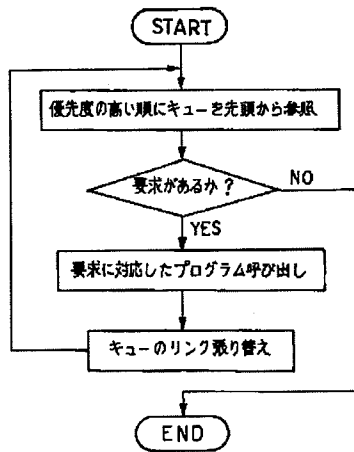
【図4】

ディスパッチャプログラムの実行する処理フローの一実施例



【図6】

従来技術の説明図



【図7】

従来技術の問題点の説明図

